SELF-DEPLOYMENT OF MOBILE AGENTS IN MANETS FOR MILITARY APPLICATIONS

Cem Şafak Şahin, Elkin Urrea*, M. Umit Uyar, Michael Conner
Bio-Inspired Computing Lab., Department of Elec. Eng., The City College of New York, NY, 10031

Ibrahim Hokelek
Applied Research Area, Telcordia Technologies, Inc., Piscataway, NJ 08854

Giorgio Bertoli and Christian Pizzo
U.S. Army Communications-Electronics RD&E Center, Fort Monmouth, NJ 07703

ABSTRACT

We present bio-inspired computation techniques, such as genetic algorithms, for real-time self-deployment of mobile agents to carry out tasks similar to military applications. Under the harsh and bandwidth limited conditions imposed by military applications, self-spreading of autonomous mobile nodes becomes much more challenging. In our approach, each mobile agent exchanges its genetic information, which is composed of speed and direction encoded in its chromosome (genome), with the neighboring nodes located in its communication range. A genetic algorithm run at the application layer as a software agent is used by each node to decide its next speed and direction among a large number of choices so that the unknown geographical area can be covered uniformly under conditions such as hostile attacks, natural (i.e., mountain, trees, lakes etc.) and man-made obstacles. We implemented a simulation software to quantify the effectiveness of the genetic algorithms under different military operational conditions (e.g., losing assets during an operation, the remaining agents should reposition themselves to compensate the lost in coverage and network connectivity). Metrics including normalized area coverage, deployment time, avoidance from obstacles over an unknown geographical area are used to demonstrate the efficiency of the self-deployment algorithm. The results show that genetic algorithms can be applied to autonomous mobile nodes and be performed as an effective tool for providing a robust solution for network area coverage under restrained communication conditions.

I. INTRODUCTION

Self-spreading of autonomous mobile nodes of a mobile ad-hoc network (MANET) over an unknown geographical area to obtain a uniform area coverage has many military applications such as search and rescue missions, mine-field clearing, and self-spreading of assets under harsh and bandwidth limited conditions. A genetic algorithm (GA) can be used by each node to select the ”fitter” speed and direction options among exponentially large number of choices converging toward a uniform node distribution.

In a typical scenario, mobile agents in an unknown terrain arrange themselves in such a way that the area covered by the nodes is maximized and/or the nodes are distributed uniformly Urrea et al. (2007b). Self-spreading autonomous mobile agents over an unknown terrain becomes more challenging under the conditions of military applications: *(i)* the geographical area (i.e., the focus of an operation) may change dramatically over time because of highly dynamic nature of tasks, *(ii)* number of nodes may decrease due to hostile attacks or malfunctions, *(iii)* nodes may become isolated since they may not have GPS or similar devices, *(iv)* communication among the neighboring mobile nodes may be forced to stop intermittently when the nodes are in a hostile environment, and *(v)* mobile agents may have to be deployed into a terrain from a single entry point as opposed to an initial (e.g., random) distribution.

Our earlier work introduced a force-based GA (*FGA*) (Sahin et al., 2008a,b; Urrea et al., 2008b) inspired by the molecular force-based distribution in physics as presented in (Heo and Varshney, 2003). In FGA, the force on each node applied by its near neighbors is used to calculate the next location and speed of the node such that the force on the node is minimized. In this paper, we study the effectiveness of FGA under a set of conditions that may be present in military applications. We consider self-spreading of mobile nodes toward a uniform distribution while *(i)* avoiding arbitrarily placed obstacles over an unknow terrain, *(ii)* loss of mobile nodes, and *(iii)* intermittent stoppage of communication.

This work has been supported by U.S. Army Communications-Electronics RD&E Center. The contents of this document represent the views of the authors and are not necessarily the official views of, or are endorsed by, the U.S. Government, Department of Defense, Department of the Army, or the U.S. Army Communications-Electronics RD&E Center.

This work has been partially supported by the National Science Foundation grants ECS-0421159 and CNS-0619577.

Report Documentation Page

Form Approved
OMB No. 0704-0188

1. REPORT DATE **DEC 2008**	2. REPORT TYPE **N/A**	3. DATES COVERED **-**

4. TITLE AND SUBTITLE **Self-Deployment Of Mobile Agents In Manets For Military Applications**	5a. CONTRACT NUMBER 5b. GRANT NUMBER 5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER 5e. TASK NUMBER 5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Bio-Inspired Computing Lab., Department of Elec. Eng., The City College of New York, NY, 10031**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S) 11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

13. SUPPLEMENTARY NOTES
See also ADM002187. Proceedings of the Army Science Conference (26th) Held in Orlando, Florida on 1-4 December 2008, The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**UU**	**8**	

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

The rest of the paper is organized as follows. The related research in the area of genetic algorithms, optimization techniques, and robotics applications are summarized in Section II. Mobility model, and an overview of FGA and related performance metrics are described in Section III. The results of simulation experiments are presented in Section IV. Section IV-B includes the concluding remarks.

II. LITERATURE REVIEW

GAs are popular in many MANET and swarm robotics applications. For example, self-deployment of mobile nodes has been studied in a variety of contexts (Heo and Varshney, 2003; Howard et al., 2002; Winfield, 2000). In (Howard et al., 2002), mobile wireless sensors deploy and organize themselves over a geographical area based on cooperative robotics using the performance metrics of uniformity of network topology, deployment time, the percentage of the region covered, and the distance traveled. (Winfield, 2000) discusses a mission scenario in which mobile robots spread into a bounded area, collect information using their sensors, and then return to their meeting point. In (Hasircioglu et al., 2008), off-line path planning for Unmanned Aerial Vehicles (UAVs) are presented. Evolutionary algorithms are used to calculate a curved path in 3-D terrain. The selection of network parameters for MANETs using GAs are proposed in (Montana and Redi, 2005).

GAs have also been popular in various distributed robotic applications. In (Chen and Zalzala, 1995), a genetic approach is presented with a distance-safety criteria for a mobile robot motion. In (Shinchi et al., 2000), the goal of autonomous robots is to move in a highway and to reach a given destination without any collision with the help of GAs. An adaptive GA is discussed in (Gesu et al., 2004) to identify targets while avoiding obstacles; the mobile robots collect information from the environment with their video cameras and light sensors, and run their own GA to stay away from static and unknown blockages and finally arrive at a given target. In (Pugh et al., 2005), performance evaluation of a noise-resistant particle swarm optimization for the unsupervised robotic learning is presented. In (Garro et al., 2006), in a path planning for the robotic applications, bio-inspired algorithms are shown to be effective to optimize the path that a robot takes to reach its assigned target. (Pugh and Martinoli, 2007) discusses the effects of unsupervised learning techniques on robotic applications where the goal is to allow robots to evolve their own controller in an automated fashion. In (Tang and Jarvis, 2003), GAs are utilized in a swarm of robots for the co-operative task of unknown environment exploration. (Byington and Bishop, 2008) focuses on the design of a distributed controller for cooperative locomotion in a swarm of robotic agents; their goal is to design an algorithm such that agents running decentralized controller are able to grasp other robots and climb over one another so that they can pass through test obstacles.

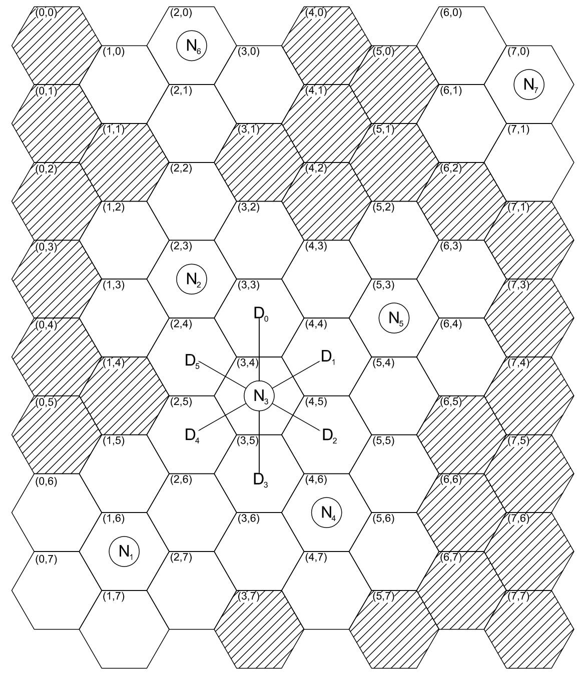

Fig. 1. An 8x8 hexagonal area partioned into logical cells, $R_{com} = 3$)

Our FGA for self-spreading autonomous mobile agents in a MANET has key differences form the above cited papers. We achieve uniform mobile node distribution by using only a limited knowledge obtained from the neighboring nodes and the local environment (location, speed, and direction) in spite of arbitrary obstacles in the terrain, mechanical malfunctions, asset losses, and intermittent stoppage of local information exchange . We assume that there is no prior knowledge of the terrain and that the mobile agents enter an unknown enemy area without any information and navigational map. Another important difference is that FGA has fully distributed intelligence without any leaders or central information distributors. In other words, FGA turns simplistic nodes with limited capabilities into an intelligent self-organized system with distributed intelligence for effective and efficient solutions for terrain independent self-spreading. Due to its operational simplicity, and flexibility, FGA can be custom-tailored for effective battery consumption or increased convergence speed (Sahin et al., 2008b).

III. SELF-SPREADING OF AUTONOMOUS MOBILE AGENTS USING GAS

A. Mobility Model and Objective

We define our geographical area as d_{max}xd_{max} divided into logical hexagonal cells. Each mobile agent has ability to move into any of the six directions with four different speeds (namely, immobile, slow, normal, and fast speeds) in the hexagonal coordinate system. The movement direction and speed for a node are determined by its FGA. As an example, Fig. 1 shows an area with 64 logical cells and seven mobile nodes, each of which can move into six directions in the hexagonal lattice (i.e., D_0 through D_5). A wireless link between two mobile nodes is represented by a vector whose dimension is in terms of hexagonal layers (one layer is equal to the center-to-center distance between two neighboring cells). In general, the wireless link state between a mobile node in location $(0,0)$ and another in location (x,y) is represented as $< x-0, y-0 > = < x, y >$. For example, in Fig. 1, the vector representing wireless link between a mobile node N_3 in location $(3,4)$ and node N_5 located in $(5,3)$ is given as $< -2, 1 >$. Suppose R is the center-to-center distance between any two mobile agents. A wireless link between these nodes $< x, y >$ is called *available*, if and only if these two nodes can communicate with each other; otherwise the link is said to be *unavailable* (Urrea et al., 2008a). The available wireless link implies $0 \leq x, y \leq R_{com} \Longleftrightarrow R \leq R_{com}$ where R_{com} is called *communication range*. The number of neighbors for a mobile agent is calculated from the total number of nodes located in its communication range. For example, the total number of neighbors for N_3 in Fig. 1 is four for $R_{com} = 3$ (i.e., N_3 can communicate with N_1, N_2, N_4, and N_5). Assuming that all mobile nodes have the same communication range, $R_{com} = 3$, the hexagonal cells which are not located within the communication range of any mobile agents are shown as gray in Fig. 1. In other words, $(1 - \frac{26}{64})$x$100 = 59\%$ of the geographical area in Fig. 1 is located within at least one mobile agent's communication range.

B. Uniform Node Distribution

Our main target is to keep the network fully connected among the mobile agents while covering a given geographical terrain uniformly under realistic conditions such as arbitrary obstacles in the terrain, stoppages due to malfunctions and hostile attacks toward one or more mobile nodes (i.e., either isolated or concentrated losses). FGA aims to provide each node with a near-optimal number of neighbors so that FGA can reach its target with the least possible number of nodes.

GAs are a member of evolutionary algorithms and work on a population of solutions set instead of dealing with a single solution. GAs have been shown to be an effective tool for providing heuristic solutions to many NP-hard problems. Typically, a GA requires two metrics to be defined: *(i)* a representation of the problem space including a solution set, and *(ii)* a fitness function in order to evaluate the goodness of each candidate solution with respect to the given goal of a GA. *Tournament* is one of the most popular selection methodologies utilized in GAs. A classical tournament is run between two individuals which may randomly be chosen from the population space. When these individuals are compared regarding to their fitness values, the one with better fitness is assigned as winner, and is permitted to reproduce. *Crossover* is applied to a chosen couple of chromosomes during reproduction to generate two new offspring. There are different types of crossover approaches in the GA implementations including single-point, two-point, cut and slice, and value-encoding crossovers. *Mutation* changes the order of the genes within a chromosome or a gene's value. Mutation prevents the GAs from getting stuck at a local optimum.

C. Forced-based Genetic Algorithms (FGA)

In our earlier work, we introduced a force-based GA (*FGA*) (Sahin et al., 2008a,b; Urrea et al., 2008b) inspired by the molecular force-based distribution in physics (Heo and Varshney, 2003). Each node is applied a force by its near neighbors, which should be summed up to zero at the equilibrium. If not, we can use this force to calculate the next location and speed of the node such that the force on the node is minimized. We should note that, although inspired by the approach of (Heo and Varshney, 2003), FGA is fundamentally different than the deterministic algorithm given in (Heo and Varshney, 2003), where the mobile nodes are pre-distributed (up to 90% prior area coverage), do not have the capability to change their speeds, and are controlled by a centralized intelligence source. In FGA, all mobile agents enter the unknown terrain from the same point (i.e., less than 10% initial area coverage), they have the capability of moving with four different speeds based on the local conditions, and the intelligence is completely distributed. These differences make FGA a much more realistic approach for military applications.

There are three main objectives for achieving the optimum self-spreading of mobile agents. The first is to have a fully connected network. The second objective is to maximize the area occupied by mobile agents while minimizing the intersection between mobile nodes' communication coverage. The third and last objective is to provide an optimum number of neighbors for each node depending on the network density. We showed that this near optimal autonomous mobile node deployment is possible if a mobile agent maintains its node degree around its analytical mean, which is called as the mean node degree $\overline{N}$ (Urrea et al., 2008a). It is used by FGA during fitness calculation.

A mobile agent collects information from its neighboring nodes about their speed, direction, location and runs the GA to optimize its own direction and speed (Sahin et al., 2008a; Dogan et al., 2008; Urrea et al., 2007a,b). In our FGA implementation, a mobile agent uses the total force applied to it by the neighboring nodes located in its communication range to decide next direction and speed. The force between two nodes is a function of the distance between them and the number of other nodes located within their communication range. After collecting local information, each agent runs its own FGA to generate several chromosomes representing candidate solutions for the next generation. These candidates are ordered according to their absolute fitness values from low to high (Sahin et al., 2008a,b), where low fitness indicates better solution. The software agent in each node runs FGA for g generations in order to populate p chromosomes at each population. At the last generation, the speed and direction which is encoded in the "fittest" chromosome are adapted by the mobile agent to be used as the input parameters of its next movement. The total force applied to a mobile node n is found as follows:

$$F(n) = \sum_{i=0}^{k} \sum_{j=0}^{k} \overline{N} \cdot (R_{com} - \mid (x - x_i) + (y - y_j) \mid) \quad (1)$$

where $\overline{N}$ is the mean node degree, k is the number of neighbors, (x, y) is the current coordinate value for the node, n is the node ID, and (x_i, y_j) is the location of a neighbor node.

D. Performance Metrics

- *Normalized Area Coverage (NAC)*: As one of the most important metrics for self-spreading algorithms, NAC value shows the portion of the geographical terrain which can be located within at least one mobile agent's communication range. The goal of our FGA is to obtain the highest possible NAC value in spite of unknown obstacles, hostile attacks, malfunctions, and silent mode.

- *Deployment Time*: This metric shows the total time it takes for the mobile nodes to converge toward a uniform distribution over a geographical area. Deployment time includes communication overhead, FGA processing time, and moving from one location to another. This metric is essential to understand the network recovery time (i.e., obtaining the best NAC value) after loss of nodes due to attacks or malfunctions, or following a silence mode.

IV. SIMULATION EXPERIMENTS

A. Software Tool and Applications

We implemented a simulation software where the mobile agents are modeled using MASON, a discrete-event multi-agent simulation tool developed in Java (Luke et al., 2005). In our software, a user is able to assign different values for the following input parameters:

- N: Total number of mobile nodes;
- R_{com} : Communication range;
- T_{max} : Maximum number of iterations;
- $\overline{N}$: Mean number of available links;
- d_{max} : Size of the geographical terrain;
- Initial node distribution;
- Number and position of obstacles.

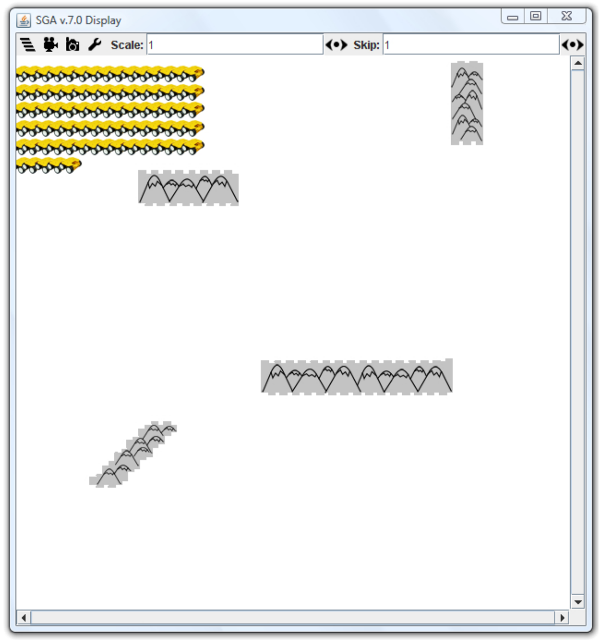

Fig. 2. Initial deployment of 80 nodes at $T = 0$

For the simlutation experiments, we consider 80 mobile agents with the same initial node distribution over an unknown region. Each mobile agent has a limited communication range (R_{com}), and, hence, can only be aware of its neighbors and the obstacles located in the node's sensing and communication range. The initial mobile agent deployment and the positions of the obstacles are shown in Fig. 2. We assume that it is not possible for the nodes to communicate through the obstacles.

We implemented FGA such that each mobile agent's movement is only affected by its current status of neighboring nodes. Due to this flexible implementation, we expect that each agent will be adaptive to the environment changes such as node failures, various terrain shapes and obstacles, and hostile attacks. To evaluate the performance and effectiveness of our FGA algorithm, we consider two types of applications. In the first application, the mobile agents are deployed in a hostile region where some of the nodes are disabled during and after the deployment. In this application, the nodes are lost either due to equipment malfunctions (i.e., isolated losses) or hostile attacks (i.e., concentrated losses). The nodes affected by either malfunctions and hostile activity are considered to be disabled for the rest of that simulation experiment. After these losses, the remaining nodes must reconfigure their positions to compensate the missing area coverage due to lost team members.

In the second application, mobile agents intentionally stop communicating with the neighboring nodes located within R_{com} distance for short periods of time. This application, called the *silence mode*, simulates the conditions where the nodes need to go undetected by hostile forces. During the silent mode, the nodes cannot communicate with each other, and therefore, cannot modify their speed and direction. We assume that, during the silent mode, the nodes keep their direction and speed that they had before they enter the silent mode. Since their speed and direction remains uncorrected by FGA, we expect that the NAC for will suffer during the silent mode. At the end of the silent mode, the nodes resume communications again.

B. Experiment Results

Fig. 3 shows the area coverage for a terrain which has arbitrary obstacles after 400 steps (i.e., iterations) of FGA. At this point of the experiment, there are three nodes that are disabled doe to malfunctions (indicated by small solid circles in Fig. 3). We can observe that, in spite of these obstacles, the mobile nodes using FGA obtain an almost uniform coverage of the area during the first 400 steps of the first application.

At step $T = 401$, the first hostile attack takes place and destroys three mobile nodes, as shown in Fig. 4. The green square at the south-east corner of the region in Fig. 4 represents the region where the enemy attacks take place. In addition, one more mobile agent experiences malfunction, reducing the total number of mobile nodes to $N = 73$.

Fig. 5 shows the mobile node deployment at $T = 600$. Between $T = 401$ and $T = 600$, another mobile node becomes disabled due to malfunction. At this point, the number of remaining mobile nodes is $N = 73$ of which

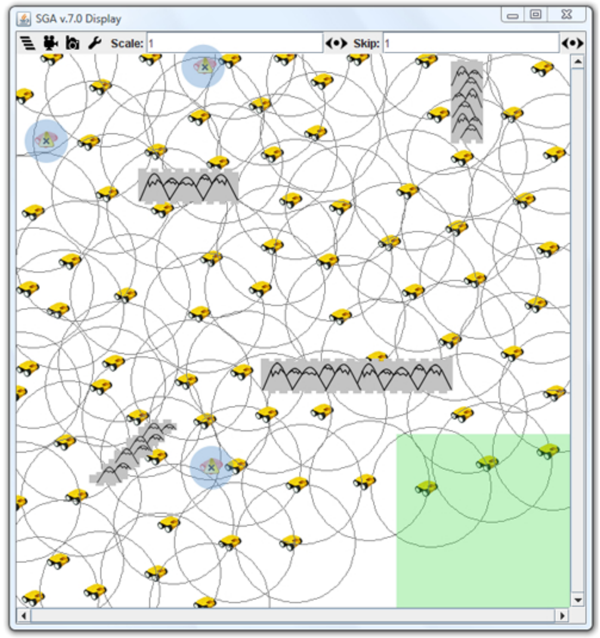

Fig. 3. Mobile node distribution for Application 1 at $T = 400$ ($N = 77$ after three disabled nodes)

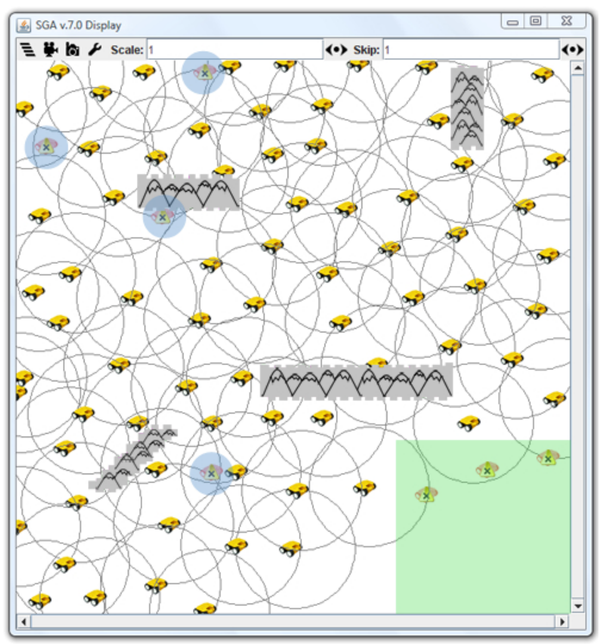

Fig. 4. Mobile node distribution for Application 1 at $T = 401$ after the first enemy attack ($N = 73$ after four disabled and three destroyed nodes)

two nodes are in the hostile region.

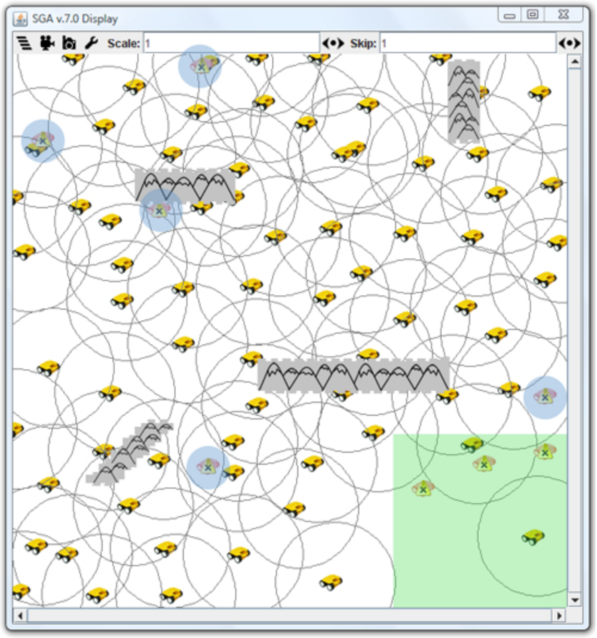

Fig. 5. Mobile node distribution for Application 1 at $T = 600$ before the second enemy attack ($N = 72$ after five disabled and three destroyed nodes)

At $T = 601$, the second enemy attack takes place destroying the two nodes in the hostile region while another node becomes disabled due to equipment malfunction, reducing the number of nodes in the experiment to $N = 69$. Fig. 6 shows the screen shot over the geographical area after the second enemy attack, which illustrates that the remaining mobile nodes keep performing FGA, and readjust their positions for a uniform area coverage.

The final mobile node distribution after running FGA for $T = 1000$ steps is presented in Fig. 7, where the remaining nodes readjust their positions to compensate for the missing nodes. At this point two more nodes are disabled bringing the total of disabled nodes due to equipment malfunction to eight and total of destroyed nodes due hostile attacks to five ($N = 67$). The network is considered fully connected at this point since all the nodes in the network are reachable by others through either one-hop or multi-hop communication.

Fig. 8 shows the convergence of FGA in terms of NAC through the iterations. The blue line in Fig. 8 illustrates that mobile nodes using FGA successfully deploy themselves around the obstacles if there were no hostile activity in the area, achieving a NAC value of 99% at $T = 1000$. Meanwhile, the red line represents the NAC when the nodes undergo malfunctions and hostile attacks. We can observe that the mobile nodes cover

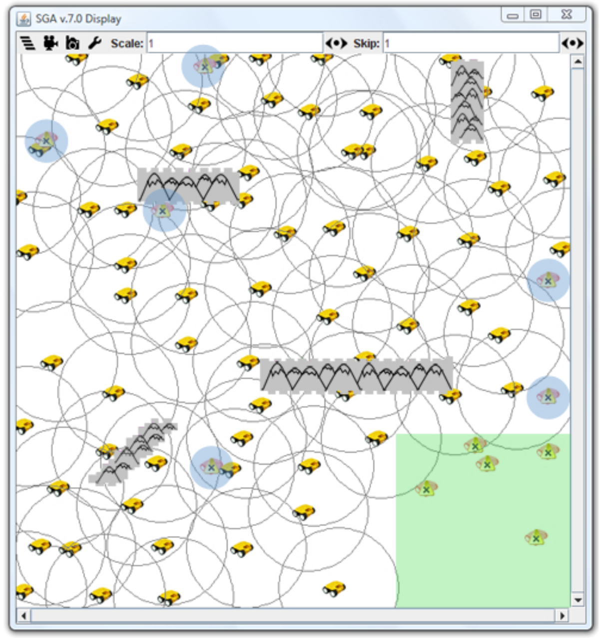

Fig. 6. Mobile node distribution for Application 1 at $T = 601$ after the second enemy attack ($N = 69$ after six disabled and five destroyed nodes)

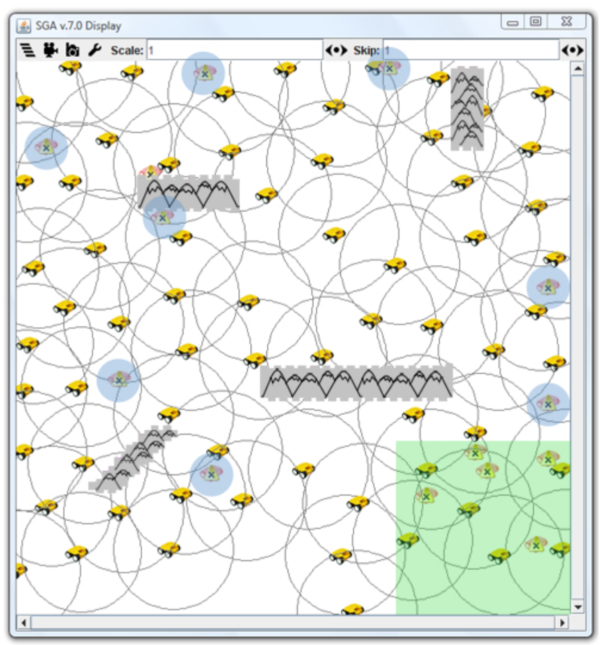

Fig. 7. Final mobile node distribution at $T = 1000$ with $n = 69$ ($N = 67$ after eight disabled and five destroyed nodes)

approximately 97% of the total area at $T = 400$.

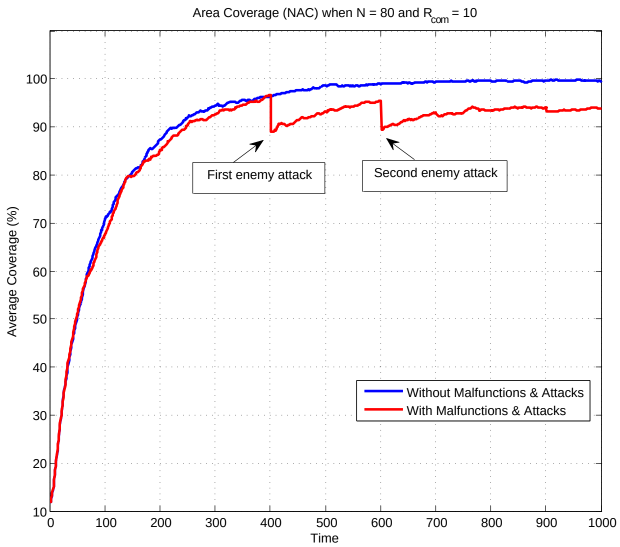

Fig. 8. Convergence of FGA in terms of NAC after $T = 1000$ steps where $N = 80$ is dropped to $N = 67$ (at $T = 401$ and $T = 601$ two enemy attacks take place while a total of eight nodes become disabled due to equipment malfunction)

After the first attack at $T = 401$, there is a drop in NAC due to the lost seven nodes, which recovers after 200 steps ($T = 600$) to the NAC value of 95% (Fig. 8). Similarly, after the second attack, there is a drop of NAC at $T = 601$, which is then compensated by the remaining nodes after they reposition themselves using FGA approximately 300 steps after the second attack ($T = 1000$).

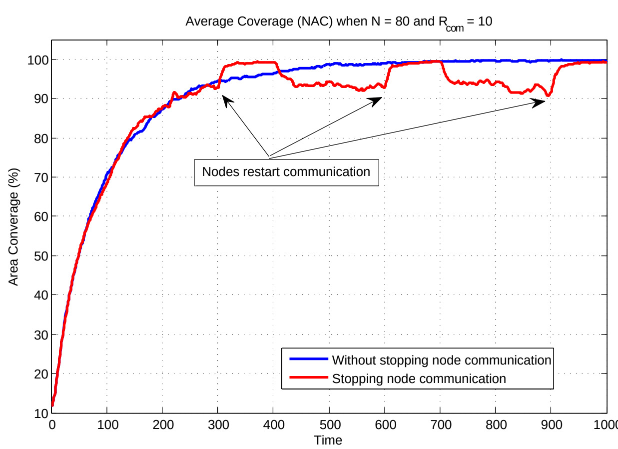

Fig. 9. Convergence of FGA in terms of NAC after 1000 steps for Application 2 (the three silent modes are $T = 100 - 300$, $T = 400 - 600$, and $T = 700 - 900$)

Fig. 9 shows the NAC for the *silence mode* application where the mobile agents intentionally stop communicating with their neighbors during short periods of time. In the experiment, the nodes perform FGA for 100 consecutive steps until $T = 100$ and then become silent for 200 iterations until $T = 300$. Similarly, the silent mode is repeated for 200 iterations between $T = 400 - 600$ and again in $T = 700 - 900$. During the silent mode, the nodes do not execute the FGA to correct their directions and speed, which results in reduced NAC values. We can observe that following silent periods, FGA significantly improves NAC values after $T = 300 - 400$, $T = 600 - 700$ and $T = 900 - 1000$.

CONCLUSIONS

In this paper, we study the effectiveness of FGA, which was introduced in (Sahin et al., 2008a; Urrea et al., 2008a; Sahin et al., 2008b) to handle harsh conditions that may be present at military applications. In this framework, the mobile nodes are deployed over an unknown territory, where there are arbitrarily placed obstacles to prevent the free movement of the nodes, and hostile activities resulting in loss of nodes and/or communication. FGA, running locally at each mobile node, adjusts the location, speed, and direction of each node through the tournament, crossover, and mutation operations defined for GAs to obtain a uniform node distribution. In FGA, the genetic information used by each node is locally obtained from its immediate neighbors. Simulation results show that FGA can be deployed successfully under conditions similar to military applications.

Future work will include the introduction of a mathematical model to formally prove the convergence of FGA.

REFERENCES

Byington, M. D. and B. E. Bishop, 2008: Cooperative robot swarm locomotion using genetic algorithms. *Proc. of the 40th Southeastern Symposium on System Theory*, 252–256.

Chen, M. and A. Zalzala, 1995: Safety considirections in the optimization of the paths for mobile robots using genetic algorithms. *1st Intl. Conf. on Genetic Algorithms in Engineering Systems: Innovations and Applications, 1995. GALESIA.*.

Dogan, C., M. U. Uyar, E. Urrea, C. Sahin, and I. Hokelek, 2008: Testbed implementation of genetic algorithms for self spreading nodes in manets. *GEM '08: Proc. 2008 Intl. Conf. on Genetic and Evolutionary Methods*, CSREA Press, USA, 10–16.

Garro, B. A., H. Sossa, and R. A. Vazquez, 2006: Path planning optimization using bio-inspired algorithms. *Proc. of IEEE Fifth Mexican Intl. Conference on Artificial Intelligence*, 319–330.

Gesu, V., B. Lenzitti, G. Bosco, and D. Tegolo, 2004: A distributed architecture for autonomous navigation of robots. *Proc. 5th IEEE Intl. Workshop on Computer Architectures for Machine Perception*.

Hasircioglu, I., H. R. Topcuoglu, and M. Ermis, 2008: 3-d path planning for the navigation of unmanned aerial vehicles by using evolutionary algorithms. *Proc. of the 10th Annual Conf. on Genetic and Evolutionary Computations*, ACM, New York, NY, 1499–1506.

Heo, N. and P. Varshney, 2003: A distributed self spreading algorithm for mobile wireless sensor networks. *Wireless Communications and Networking, 2003. WCNC 2003. 2003 IEEE*, **3**, 1597–1602 vol.3.

Howard, A., M. Mataric, and G. Sukhatme, 2002: Mobile sensor network deployment using potential fields: A distributed, scalable solution to the area coverage problem. *Proc. 6th Intl. Symp. on Distributed Autonomous Robotics Systems, Fukuoka, Japan*.

Luke, S., C. Cioffi-Revilla, L. Panait, K. Sullivan, and G. Balan, 2005: Mason: A multiagent simulation environment. *Simulation*, **81**, 517–527, doi:http://dx.doi.org/10.1177/0037549705058073.

Montana, D. and J. Redi, 2005: Optimizing parameters of a mobile ad hoc network protocol with a genetic algorithm. *GECCO '05: Proc. Conf. on Genetic and Evolutionary Computation*, ACM, New York, NY, USA, 1993–1998.

Pugh, J. and M. Martinoli, 2007: Parallel learning in heterogenous multi-robot swarms. *Proc. of IEEE 2007 Congress on Evolutionary Computation*, 3839–3846.

Pugh, J., M. Martinoli, and Y. Zhang, 2005: Particle swarm optimization for unsupervised robotic learning. *Proc. of IEEE 2005 Swarm Intelligence Conference*, 92–99.

Sahin, C., E. Urrea, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli, and C. Pizzo, 2008a: Genetic algorithms for self-spreading nodes in manets. *GECCO '08: Proc. Genetic and Evolutionary Computation Conf.*, ACM, New York, NY, USA, 1141–1142.

— 2008b: Uniform distribution of mobile agents using genetic algorithms for military applications in manets. *Proc. Intl. Conf. on Military Communications (MILCOM)*, (to appear).

Shinchi, T., M. Tabuse, T. Kitazoe, and A. Todaka, 2000: Khepera robots applied to highway autonomous mobiles. *Artificial Life and Robotics*, **7**, 118–123.

Tang, K. W. and R. A. Jarvis, 2003: Application of genetic algorithms to robotic swarm simulation. *Proc. of the 2003 Conf. on Evolutionary Computation*, 1064–1071.

Urrea, E., C. Sahin, I. Hokelek, M. U. Uyar, M. Conner, G. Bertoli, and C. Pizzo, 2008a: Bio-inspired topology control for knowledge sharing mobile agents. *Special Issue on Bio-inspired Computing and Communication in Wireless Ad Hoc and Sensor Networks. Elsevier*, (to appear).

Urrea, E., C. Sahin, M. U. Uyar, M. Conner, I. Hokelek, G. Bertoli, and C. Pizzo, 2008b: Comparative evaluation of genetic algorithms for force-based self-deployment of mobile agents in manets. *GEM '08: Proc. 2008 Intl. Conf. on Genetic and Evolutionary Methods*, CSREA Press, USA, 90–95.

Urrea, E., C. Sahin, M. U. Uyar, M. Conner, I. Hokelek, G. Bertolli, and C. Pizzo, 2007a: Simulation experiments for knowledge sharing agents using genetic algorithms in manets. *Proc. Intl. Conf. on Artificial Intelligence and Pattern Recognition*, 369–376.

— 2007b: Uniform manet node distribution for mobile agents using genetic algorithms. *Proc. Intl. Conf. on Genetic and Evolutionary Methods*, 24–30.

Winfield, A., 2000: Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots. *Distributed Autonomous Robotic Systems*, 273–282.